

# **Speech Recognition and Voice Control for Baxter Research Robot**

---

Supervisor:

Assistant Prof. Emre Uğur

Computer Engineering Department, Bogazici University

e-mail: [emre.ugur@boun.edu.tr](mailto:emre.ugur@boun.edu.tr)

Supervisee:

Bilgehan NAL

Bachelor Degree of Computer Engineering Department at

Marmara University

e-mail: [bilgehannal@marun.edu.tr](mailto:bilgehannal@marun.edu.tr) / [bilgehan1.03@gmail.com](mailto:bilgehan1.03@gmail.com)

## **Table of Contents**

1. Introduction
2. Aim
3. Technical Content
  - 3.1. Speech Recognition
  - 3.2.Communication with Baxter
  - 3.3.Commands
4. Import and Usage
  - 4.1. Import and Usage of “Vocie\_Recogniser”
  - 4.2. Usage of “Voice\_Recogniser” in the Project
5. Result
6. Source Materials

## 1. Introduction

Human–robot interaction is the study of interactions between humans and robots. It is often referred as HRI by researchers. Human–robot interaction is a multidisciplinary field with contributions from human–computer interaction, artificial intelligence, robotics, natural language understanding, design, and social sciences.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems. These speech industry players include Google, Microsoft, IBM, Baidu, Apple, Amazon, Nuance, SoundHound, IflyTek, CDAC many of which have publicized the core technology in their speech recognition systems as being based on deep learning.

## 2. Aim

In previous studies, we developed a Face System with emotions and face actions for Baxter Research Robot and we can control the situation of the Baxter's face publishing a String message to a determined topic.

The aim of the project is to control the situation of the Face System with a voice. Also developing a modular small library for converting a speech to text to use for any python project.

## 3. Technical Content

The project consists of one node. This node recognises the speech of a human, after it decides the command, it will publish a message to control the Baxter robot.

Users have to push “space key” before the speech to activate the system. Program stops listening after a command for avoiding the recognition of unexpected speech.

In addition all program is written using Python programming language.

### 3.1.Speech Recognition

Google Speech Recognition API is used for recognition of speeches. There is a function, in the project code, named “listen()”, this function returns the what the user says.

```
def listen():
```

In this function, there are some methods to display the process of the speech recognition, before returning a value, it gives a notification about the recognitions was done.

Requirements for using Google Speech Recognition:

- **Python** 2.6, 2.7, or 3.3+ (required)
- **PyAudio** 0.2.11+ (required only if you need to use microphone input, Microphone)
- **PocketSphinx** (required only if you need to use the Sphinx recognizer, recognizer\_instance.recognize\_sphinx)
- **Google API Client Library for Python** (required only if you need to use the Google Cloud Speech API, recognizer\_instance.recognize\_google\_cloud)
- **FLAC encoder** (required only if the system is not x86-based Windows/Linux/OS X)

```
string = r.recognize_google(audio, language='tr-TR')  
string = r.recognize_google(audio, language='tr-TR')
```

we could take the string from speech using recognize\_google function. In the above sample code, there is a variable named r. “r” is a recogniser object. (sr.Recogniser()).

### 3.2.Communication with Baxter

The node used for the project controls the Baxter publishing a String message to display\_chatter topic. In the project of “Emotions and Face Actions of Baxter Research Robot”, there is a node to subscribe display\_chatter topic and a user can control the face and head situation through this topic using certain commands.

After the program convert speech to a string, program decides which command would it publish using the “decideAction()”.

### 3.3.Commands

There are some certain commands to control the Baxter's head and face can be published, but the datas (came from Google Speech API) can not be those certain commands because of language. Google Speech API is used for recognising the Turkish speech. For that reason, program needs to process speech string and convert to those commands.

“decideAction()” is used for converting the speech data to the certain commands.

```
def decideAction(string):
```

it takes a string parameter and returns the processed version of that string.

There is a txt file in “data” folder named “voice\_command.txt” which is used for the commands. “voice\_command.txt” has many data about the action of the robot. Structure of the datas should be “x.y”. ‘x’ is the word that the machine expects from the people and ‘y’ is a command for Baxter. If a person say a sentence which includes ‘x’, “decideAction()” function returns ‘y’. Then command of ‘y’ is published to the determined topic.

Example data of “voice\_command.txt”

<i>mutlu.happy</i>		<i>üzül.sad</i>		<i>kızgın.angry</i>
<i>sevin.happy</i>		<i>üzgün.sad</i>		<i>öfke.angry</i>
<i>gül.happy</i>		<i>kötü.sad</i>		<i>sinir.angry</i>

## 4. Import and Usage

“Voice\_Recogniser.py” library can be used in all project written with Python. Also this library used for the project. “Voice\_Recogniser.py” is a kind of small library which helps for recognising the voice and converting the speech to text powered by Google Speech API.

### 4.1. Import and Usage of “Vocie\_Recogniser”

A programmer who wants to use “Voice\_Recogniser.py” for his/her project has to import “Vocie\_Recogniser” to his/her python script.

```
import Voice_Recogniser
```

Voice\_Recogniser is a kind of class, the programmer have to generate an object from the class to use the library in the project. Thanks to Google, “Voice\_Recogniser” allows to use 119 different language and variants.

You can see the all allowed language and variants: <https://goo.gl/GzQUBd>

There are two constructors of “Voice\_Recogniser”. One of them can be used for only converting speech to text, another one can be used for converting speech to text and processing the text according to the given txt file.

```
recogniser = Voice_Recogniser.Voice_Recogniser()  
recogniser = Voice_Recogniser.Voice_Recogniser("sample.txt")
```

After creating the object, “Voice\_Recogniser” has three functions which can be used.

These are:

```
def listen_language(self, language):  
def listen(self):  
def process(self, string):
```

“listen\_language” and “listen” function returns the text version of the speech. Programmer can determine the language using “listen\_language” function. This function takes a String parameter as a code of the languages for the Google Speech API. There are several language code was already defined in “Voice\_Recogniser”, the programmer can also use these defined variables. Default version of the “listen\_language” is “listen” function which can be also used for converting speech to text, but default version allows only one language (US English).

“process” function returns a string according to the given parameter of string and given txt file. You can see how to run “process” function in “3.3. Commands” section.

#### 4.2. Usage of “Voice\_Recogniser” in the Project

We imported the “Voice\_Recogniser” to our node, and we created an object from this library.

```
recogniser =  
Voice_Recogniser.Voice_Recogniser("/home/{}/ros_ws/src/baxter_face/scripts/data/voice_command.txt".format(getpass.get  
user()))
```

Our aim is converting to Turkish speech to text, so we are using “listen\_language” with the parameter of “recogniser.TURKEY”. Then we have to process and convert the text version of the speech to the Baxter Face Control Commands.

```
command = recogniser.process(recogniser.listen_language(recogniser.TURKEY))
```

Finally, we can publish the command to the “display\_chatter” topic as a String message.

## 5. Result

In this work, small library “Voice\_Recogniser” is written using Google Speech API to use in any Python project for recognising the speech and converting the speech to text. Then, we tried to make a voice controller for the Baxter’s face actions and situations.

## 6. Source Materials

- [1] <https://cloud.google.com/speech/docs/languages>
- [2] <https://cloud.google.com/speech/>
- [3] [https://en.wikipedia.org/wiki/Human–robot\\_interaction](https://en.wikipedia.org/wiki/Human–robot_interaction)
- [4] [https://en.wikipedia.org/wiki/Speech\\_recognition](https://en.wikipedia.org/wiki/Speech_recognition)